IN THE UNITED STATES PATENT & TRADEMARK OFFICE

Inventor(s):	KENDALL, Chad; DAVIS, Tom; NAHUM, Shay
Title:	CONFIGURABLE PACKET PROCESSOR
Serial No.:	09/988,939
Filed:	21 November 2001
Examiner:	MATTIS, Jason E. Art Unit: 2665

AFFIDAVIT

1. I, Shay NAHUM, together with Chad KENDALL and Tom DAVIS, am the co-inventor of the invention claimed in this U.S. Patent Application No. 09/988,939 ("the patent application"). As such, I have personal knowledge of the facts stated in this affidavit except where stated to be made on the basis of information and belief in which case I verily believe them to be true.

2. Chad KENDALL, Tom DAVIS and I conceived of configurable packet processors and related methods as claimed in the patent application prior to 9 July 2001.

3. Exhibit A to this affidavit is a copy of portions of a confidential invention disclosure document we prepared and provided to our company's in-house patent department describing technical information about our configurable packet processors. The version of the disclosure document in Exhibit A was completed on or before 14 June 2001.

4. The disclosure document describes a configurable packet processor using three memories. Specifically, the disclosure document describes an internally stored parser memory (e.g. RAM), an internally stored match engine memory (e.g. content addressable memory) and an externally stored context memory (e.g. 1 million entry capacity).

5. I have reviewed claims 1 through 28 of the patent application. A copy of the claims, as I have reviewed them, is set out in Exhibit B to this affidavit.

6. The disclosure document, particularly at page 4 and Figure 1, describes all the steps claimed in each of method claims 1 through 10.

7. The disclosure document, particularly at page 4 and Figure 1, describes all the features claimed in each of apparatus claims 12 through 28.

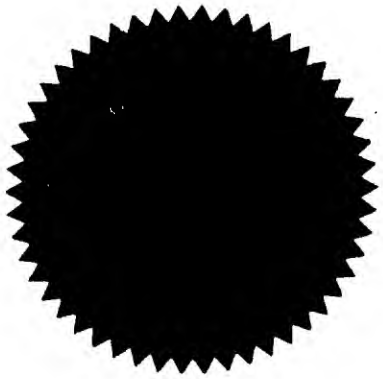
SWORN BEFORE ME at the City of Ottawa, in the Province of Ontario, Canada, this 3rd day of February, 2006.

A Notary Public

Shai NAHUM

THIS IS EXHIBIT "A" TO THE AFFIDAVIT OF
Shay NAHUM
SWORN BEFORE ME THIS
25th day of February, 2006

A Notary Public



Technical Information

Local reference:
Alcatel reference: 137425
Short Title Efficient Configurable Packet Processing in Hardware

Technical Field of the Problem:

Any device which is tasked with parsing through a protocol stack to determine what actions to perform on that packet. Such devices could include network processors, ASICs, or programmable logic devices.

Closest Prior Art Solutions:

Many solutions exist for performing packet processing - from hardwired ASICs to fully programmable network processors. Hardwired ASICs can handle very large packet rates for packets with a specific protocol that is well known. Network processors are very flexible, but lack in performance. Therefore, solutions have been developed which allow some flexibility in a hardware-based device. These solutions provide higher performance than network processors, but are more flexible than hardwired ASICs.

The closest prior art solutions to the invention use a tree-search methodology to determine the actions to perform on a packet. Starting from the first bits of a packet, bit segments are extracted and used to access a memory. Each memory access indicates which next bit segment to access and may also provide some further action to take. The last memory access will determine the final action to take on the packet. The final action could include whether or not to discard the packet, where to forward the packet, what quality of service to assign to the packet, etc.

What problems exist with the prior art solutions, which the invention solves?

The nature of the packet processing problem requires many memory accesses to arrive at the final packet actions as there can be many protocols in the stack of one packet. In prior art solutions, each bit field is used to access memory directly (i.e. as an address) so that for sparse protocols, large memories are required. These large memories often must be stored in memory external to the device, resulting in performance limitations because of the limited number of accesses that can be made in each packet time.

The invention overcomes these problems by using internal CAM (content addressable memory) and RAM memories for sparse protocols (such as PPP). Instead of providing 64k of memory for the 16bit PPP protocol when only a few values are important, the values are stored internally in a CAM. Only protocols which are densely populated are stored in external memory. This allows a significant performance improvement for many common protocol stacks as internal memories can be accessed much more frequently than external memories. In addition, the next leaf information is not stored for each leaf of the tree, further reducing memory requirements.

Name of person filling out the form: Chad Kendall
Date: 2001-06-13 June 14, 2001

Brief description of the solution, pointing out what is different from prior art. Attach a drawing if it helps understanding.

Please refer to Figure 1 for the following explanation of the invention.

The invention uses three memories. The first memory is a 128 entry Parser RAM (PR) stored internally. As shown in Figure 1, the Parser RAM contains the lengths and offsets of a label and a protocol to extract from the packet. It also contains a Match Engine index, a total offset, and a context base address. The total offset is used to increment the packet stake when the iteration is completed. All protocols and labels are extracted relative to the current packet stake.

The second memory is a 128 entry Match Engine (ME). The Match Engine acts like a ternary CAM and is also stored internally. It is provided with a 'key' which is searched for within it's 128 entries. The 'key' consists of the Match Engine index appended with the protocol. That Match Engine index is simply used to distinguish between different protocols.

The third memory is a 1 million entry Context memory and is stored externally (shown in blue). It contains context associated with the packet as well as a Match Engine index.

Protocol processing begins with a Parser RAM index obtained from an ATM or POS channel lookup. The Parser RAM index is simply the address into the Parser RAM. If the label length in the Parser RAM is non-zero, the label extracted is added to the context base address and used to access the external context memory. Context memory returns information associated with the label and a Match Engine index. The Match Engine index is prepended with the protocol and used to access the Match Engine. A match results in an action being returned which could include the action to extract another protocol from the packet. In this case, a Parser RAM index is returned as well. The Parser RAM index identifies the entry in the Parser RAM used to extract the next protocol and the process repeats. The process ends when the Match Engine returns a final action for the packet.

To allow the label lookup in Context to alter the action of the packet, any time a label lookup occurs between a protocol extraction and a Match Engine search, the Match Engine index from Context is used over the Match Engine index from the Parser RAM.

Because a pipelined hardware solution is used, only a fixed sequence of operations is allowed. The sequence is:

1. Parser RAM extraction 0, Match Engine 0 search
2. Parser RAM extraction 1, Context lookup, Match Engine 1 search
3. Parser RAM extraction 2, Match Engine 2 search
4. Parser RAM extraction 3, Match Engine 3 search
5. Parser RAM extraction 4, Context lookup, Match Engine 4 search
6. Parser RAM extraction 5, Match Engine 5 search

Any step in the above sequence is optional and can be skipped. Any packet which cannot be handled with this sequence of operations creates an exception.

Name of person filling out the form: Chad Kendall
Date: 2001-06-13 June 14, 2001

Basic Idea which lead the inventor to the above solution:

The desire for a programmable hardware solution which could process packets at a high rate with limited external memory bandwidth led to the idea of utilizing internal CAM and RAM memories for most protocols.

Advantage(s) with respect to the best prior art solution (quantified if possible):

The main advantages of the invention are that less external memory bandwidth is required and that less memory overall is required. The first advantage permits a programmable solution that can process packets at a high rate. The second advantages reduces the cost of the solution.

Disadvantage(s) or drawbacks of the solution of the invention (quantified if possible):

The invention is based upon a pipelined hardware approach and therefore new protocols might not fit into the solution. In addition, the invention does not efficiently support nested protocols where the inner protocol values depend upon the outer protocol value. This is due to the limited number of Parser RAM entries which contain the associated context base address.

Main Figure(s):

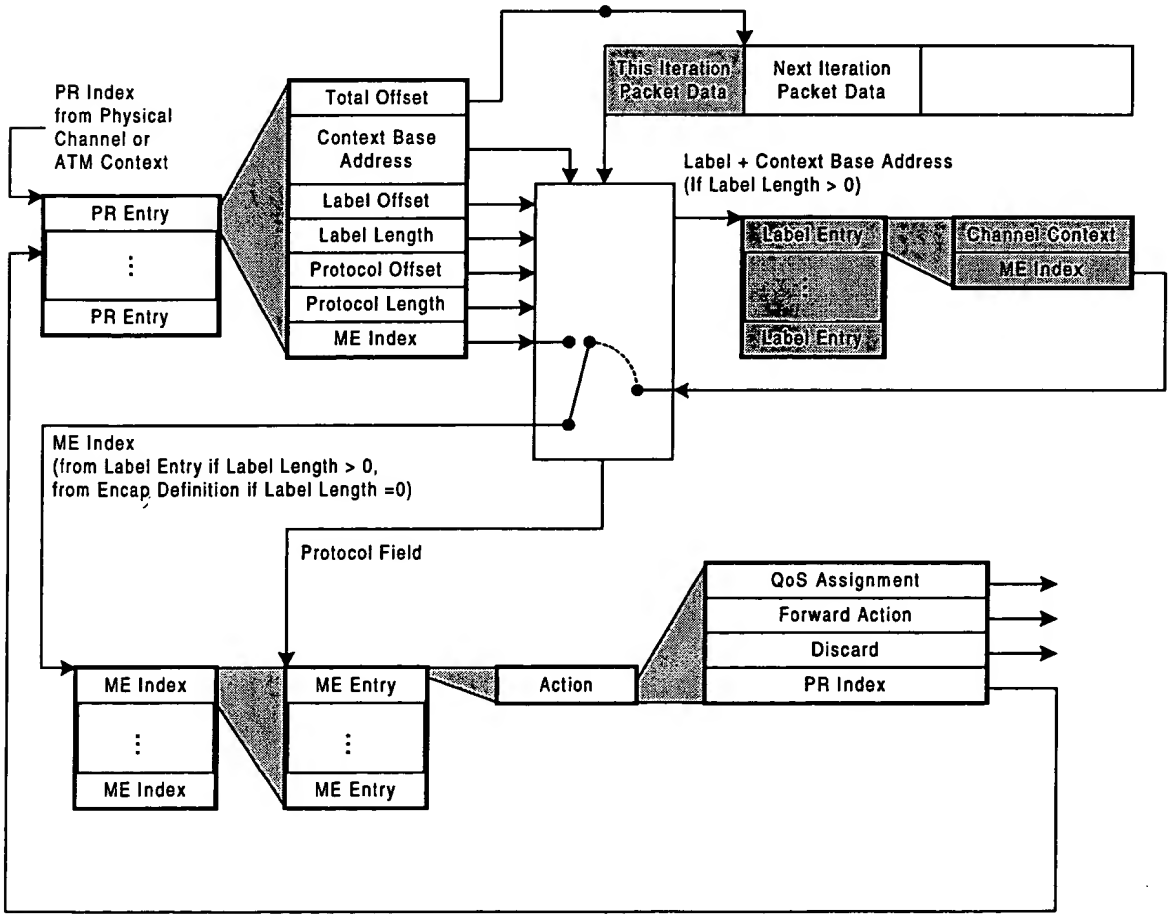


Figure 1 :Diagram of Hardware Packet Processor

Name of person filling out the form: Chad Kendall
Date: 2001-06-13 June 14, 2001

Commercial value (possible use by competitors):
Any system that requires packet processing (almost all switches and routers) that is operating above a certain rate (currently estimated at over OC48 rates).

How can infringement be detected:
Any product which performs packet processing at a high rate (i.e. OC-48 and above) and advertises the ability to upgrade the product to support new protocols without the addition of new hardware could be using the invention. The amount of memory required for their packet processing device could be an indication as well as the specification for how new protocols can be supported.

Remarks:

Name of person filling out the form: Chad Kendall
Date: 2001-06-13 June 14, 2001

Local reference:
Alcatel reference: 137425
Short Title Efficient Configurable Packet Processing in Hardware

Proposal of independent claims:

1. A device for processing packets in which a plurality of protocols are supported, characterized in that,

the device contains first and second internal memories,

the first internal memory provides, based on a channel on which a packet is received, address information to access an external memory containing context information associated with the packet, and

the second internal memory provides, based on matching label and index information obtained from the external memory, an action to be executed with respect to the packet.

2. A device as claimed in claim 1, wherein the action returned is to extract another protocol from the packet and the second internal memory provides an index into the first internal memory used to extract the next protocol.

3. A device as claimed in claim 2, wherein the first internal memory is ram and the second internal memory is a CAM.

Name of person filling out the form: Chad Kendall
Date: 2001-06-13 June 14, 2001

THIS IS EXHIBIT "B" TO THE AFFIDAVIT OF

Shay NAHUM

SWORN BEFORE ME THIS

3rd day of February 2006.

A Notary Public

WHAT IS CLAIMED IS:

1. A method for packet processing comprising,
   a) obtaining first information regarding a packet;
   b) using the first information as an index into a parser memory;
   c) retrieving from the parser memory an entry comprising a location in the packet of one or more protocol bits specifying a protocol associated with the packet;
   d) obtaining a match engine index; and,
   e) using the protocol bits and the match engine index as a key to retrieve a match engine entry from a match engine memory, the match engine entry comprising an action to take on the packet.

2. The method of claim 1 wherein the match engine index is included in the parser memory entry.

3. The method of claim 1 wherein the parser memory entry comprises a context memory base address and either a location in the packet of a set of label bits or an indication that there are no label bits,

wherein, if the parser memory entry includes a location of a set of label bits, the method comprises retrieving from the packet the label bits, and obtaining the match engine index comprises using the context memory base address and label bits to retrieve from a context memory an entry comprising the match engine index.

4. The method of claim 3 wherein, if the location in the packet of a set of label bits indicates that there are no label bits, obtaining the

match engine index comprises retrieving a match engine index included in the parser memory entry.

5. The method of claim 4 wherein the match engine memory comprises a content-addressable memory.

6. The method of claim 4 wherein the match engine memory comprises a ternary content-addressable memory.

7. The method of claim 1 wherein the match engine memory comprises a content-addressable memory.

8. The method of claim 1 wherein obtaining the first information comprises identifying a channel with which the packet is associated.

9. A method for packet processing in a packet processing system, the method comprising:

a step for obtaining first information regarding a packet;

a step for retrieving an entry corresponding to the first information from a parser memory;

a step for retrieving from the packet one or more protocol bits identified by the parser memory entry;

a step for retrieving from a match engine memory a match engine memory entry comprising an action to perform using a match engine key comprising a combination of the protocol bits and a match engine index; and,

a step for performing the action specified in the retrieved match engine entry.

10. The method of claim 9 wherein the action comprises extracting information relating to another protocol from the packet.

11. The method of claim 9 wherein the action is selected from the group consisting of forwarding the packet, discarding the packet, adding additional header information to the packet, associating the packet with a quality of service level, and extracting information relating to another protocol from the packet.

12. A packet processing apparatus comprising:

a control logic circuit;

a parser memory accessible to the control logic circuit the parser memory comprising a plurality of entries each specifying a location in a packet of one or more protocol bits and at least some of which specifying a match engine index;

a match engine memory accessible to the control logic circuit, the match engine memory comprising a plurality of entries each specifying an action to be taken; and,

a context memory accessible to the control logic circuit, the context memory comprising a plurality of entries each specifying a match engine index;

wherein the control logic circuit is configured to generate a match engine key by combining protocol bits of a packet identified in a parser memory entry with a match engine index from an entry of either the parser memory or the context memory, to retrieve from the match engine memory an entry corresponding to the match engine key, and to perform an action specified in the match engine entry.

13. The apparatus of claim 12 wherein the control logic circuit comprises an integrated circuit and the parser memory is integrated with the control logic circuit.

14. The apparatus of claim 12 wherein the control logic circuit comprises an integrated circuit and the match engine memory is integrated with the control logic circuit.

15. The apparatus of claim 14 wherein the control logic circuit comprises an integrated circuit and the parser memory is integrated with the control logic circuit.

16. The apparatus of claim 15 wherein the context memory is external to the control logic circuit and the control logic circuit comprises an integrated interface to the context memory.

17. The apparatus of claim 16 wherein the parser memory comprises 512 or fewer entries.

18. The apparatus of claim 17 wherein the match engine memory comprises 512 or fewer entries.

19. The apparatus of claim 16 wherein the control logic circuit comprises a pipelined architecture.

20. A configurable device for processing packets, the device supporting a plurality of protocols, the device comprising:

a first internal memory comprising a plurality of entries;

a second internal memory comprising a plurality of entries each comprising an action to be taken on the packet;

logic circuitry for identifying a channel value associated with the packet, retrieving an entry from the first memory using the channel value as an index, and obtaining from the entry address information identifying a set of entries in an external context memory applicable to the channel value;

logic circuitry for using the address information and one or more bit values from the packet to retrieve from the external context memory one entry from the set of entries; and,

logic circuitry for using information from the one entry retrieved from the external context memory to retrieve from the second memory an action to be taken on the packet.

21. The device of claim 20 wherein the action to be taken on the packet comprises extracting information relating to a protocol from the packet.

22. The device of claim 21 wherein the second memory comprises a content addressable memory.

23. The device of claim 22 wherein the first memory comprises a random access memory.

24. A packet processing device comprising:

means for retrieving first information about a received packet;

means for retrieving an entry corresponding to the first information, the entry comprising a location in the packet of one or more protocol bits specifying a protocol associated with the packet and a match engine index;

means for generating a match engine key; and,

means for retrieving an action corresponding to one of a plurality of match engine entries which matches the match engine key; and,

means for performing the action.

25. The packet processing device of claim 24 wherein the first information comprises an ATM channel associated with the packet.

26. The packet processing device of claim 25 wherein the means for performing the action included means for forwarding the packet to another packet processing device.

27. The packet processing device of claim 24 comprising

means for determining from the entry whether to retrieve an entry from an external context memory and, means for retrieving an entry from the external context memory, wherein the means for generating the match engine key is adapted to generate the match engine key using information in the entry from the external context memory.

28. The packet processing device of claim 27 wherein the means for generating a match engine key, the means for retrieving an action and the means for retrieving an entry from the external context memory are incorporated on a single integrated circuit.